

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-116251

(43)Date of publication of application : 06.05.1998

(51)Int.Cl. G06F 15/00
G06F 9/06

(21)Application number : 09-187270

(71)Applicant : SUN MICROSYST INC

(22)Date of filing : 30.06.1997

(72)Inventor : WALLS DOUGLAS
WALKER KATHRYN
MORI ROBERT F

(30)Priority

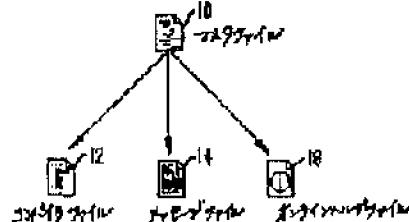
Priority number : 96 680499 Priority date : 01.07.1996 Priority country : US

(54) METHOD, DEVICE, AND SYSTEM FOR MESSAGE MANAGEMENT, AND OPTION MANAGING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To hold consistency among various documentation components by generating a message catalog file used to localize a message text.

SOLUTION: The output of a master message file 10 is constituted by using one set of tools, which are implemented as codes stored on various computer usable storage media of a computer system. Respective output files are generated by the respective tools when a compiler for generating the latest version of a file generated from a master message file 10 is integrated and built. Namely, a PREPERR tool operates the master message file 10 in a 1st process step to generate a header (.h) file 12, an on-line help source (.htg) file 16, and a source (.msg) file 14 used to generate a message catalog.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-116251

(43)公開日 平成10年(1998)5月6日

(51)Int.Cl.⁸

G 0 6 F 15/00
9/06

識別記号

3 1 0
5 3 0

F I

G 0 6 F 15/00
9/06

3 1 0 Z
5 3 0 T

審査請求 未請求 請求項の数12 FD (全 15 頁)

(21)出願番号 特願平9-187270

(22)出願日 平成9年(1997)6月30日

(31)優先権主張番号 08/680,499

(32)優先日 1996年7月1日

(33)優先権主張国 米国(US)

(71)出願人 591064003

サン・マイクロシステムズ・インコーポレーテッド

SUN MICROSYSTEMS, INCORPORATED

アメリカ合衆国 94303 カリフォルニア州・パロ アルト・サン アントニオ ロード・901

(72)発明者 ダグラス・ウォールズ

アメリカ合衆国・94002・カリフォルニア州・ペルモント・ラッセン ドライブ・1009

(74)代理人 弁理士 山川 政樹

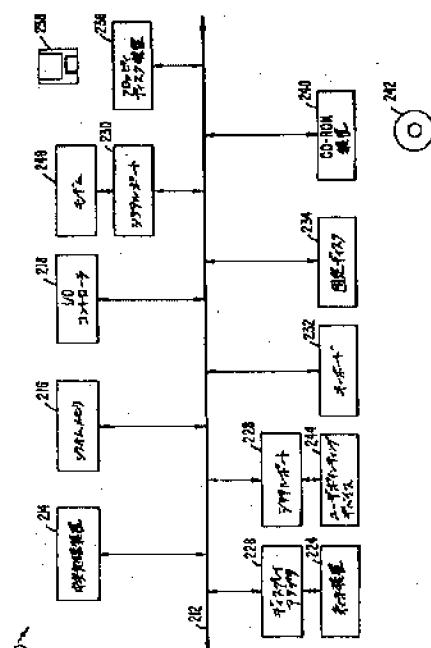
最終頁に続く

(54)【発明の名称】 メッセージ管理方法および管理装置ならびに管理システムならびにオプション管理方法

(57)【要約】

【課題】 ソフトウェアシステムにおけるドキュメンテーションの変更およびローカライズの際に、種々のドキュメンテーションコンポーネント間の一貫性を保つ。

【解決手段】 単一のマスタメッセージファイルからメッセージを生成し、メッセージカタログをローカライズし、オンラインヘルプメッセージを表示するために必要なファイルを生成する。メッセージシステムの改訂及び更新を、ただ一つのファイル、すなわちマスタメッセージファイルを編集することによって行う。マスタメッセージファイルを処理して出力を構築するためのツールを設けた。



【特許請求の範囲】

【請求項1】全てのメッセージ用のテキストを保持し、1カ所で情報をサポートすると共に、各々シンボリックハンドル、メッセージテキスト、及びヘルプメッセージテキストを含む複数のメッセージエントリを有する单一のマスタメッセージファイルに基づくコンピュータ実装されるメッセージ管理方法において：該マスタメッセージファイル中の情報に基づいて、オンラインヘルプを生成するために用いられるヘルプボリュームを生成するステップと；該マスタメッセージファイル中の情報に基づいて、メッセージを生成するために用いられるソースファイルに入れられるヘッダファイルを生成するステップと；該マスタメッセージファイル中の該情報に基づいて、メッセージテキストをローカライズするために用いられるメッセージカタログファイルを生成するステップと；を具備したメッセージ管理方法。

【請求項2】全てのメッセージ用のテキストを保持し、1カ所で情報をサポートすると共に、各々シンボリックハンドル、メッセージテキスト、及びヘルプメッセージテキストを含む複数のメッセージエントリを有する单一のマスタメッセージファイルに基づくコンピュータ実装されるメッセージ管理方法において：該マスタメッセージファイルからコンパイラヘッダファイルを生成するステップと；表示しようとする必要メッセージ情報のシンボリックハンドルを設けるステップと；該シンボリックハンドル及び該ヘッダファイルを処理して該必要情報にアクセスするステップと；該ヘッダファイルからアクセスされた必要メッセージ情報を表示するステップと；該マスタメッセージファイルから各々ヘルプメッセージ及びヘルプメッセージIDを含む複数のエントリを有するヘルプボリュームを生成するステップと；該アレイからアクセスされた該必要メッセージ情報に含まれるアクセスされたシンボリックハンドルを用いて、該必要メッセージ情報と関連した必要ヘルプメッセージテキストを保持する該ヘルプボリューム中のエントリを指定するステップと；ユーザ入力に応答して該必要ヘルプメッセージテキストを選択的に表示するステップと；を具備したメッセージ管理方法。

【請求項3】全てのメッセージ用のテキストを保持し、1カ所で情報をサポートすると共に、各々シンボリックハンドル、メッセージテキスト、及びヘルプメッセージテキストを含む複数のメッセージエントリを有する单一のマスタメッセージファイルに基づくコンピュータ実装されるメッセージ管理方法において：該マスタメッセージファイルからコンパイラヘッダファイルを生成するステップで、該コンパイラヘッダファイルが、マスタメッセージファイル中のエントリのシンボリックハンドルを列挙して整数として機能する列挙シンボリックハンドルを生成するた列挙ヘッダファイル、及び列挙シンボリックハンドルによって指示されるアレイ要素を有するメ

ッセージアレイファイルを含み、該アレイ要素の各々がアレイをアクセスするために用いられるシンボリックハンドルを含むマスタメッセージファイルエントリからの情報を含んでいるステップと；該列挙ヘッダファイルを用いて、表示しようとする必要メッセージ情報の指示インデックス用列挙シンボリックハンドルを生成するステップと；該指示インデックス用列挙シンボリックハンドルを用いて該メッセージアレイファイル中の該必要メッセージ情報を含む指定されたアレイ要素にアクセスするステップと；該指定されたアレイ要素に含まれる必要メッセージ情報を表示するステップと；各々ヘルプメッセージ及びヘルプメッセージIDを含む複数のエントリを有するヘルプボリュームを該マスタメッセージファイルから生成するステップと；該指定されたアレイ要素に含まれるシンボリックハンドルを用いて、該必要メッセージ情報と関連した必要ヘルプメッセージテキストを保持する該ヘルプボリューム中のエントリを指定するステップと；ユーザ入力に応答して該必要ヘルプメッセージテキストを選択的に表示するステップと；を具備したメッセージ管理方法。

【請求項4】該マスタメッセージファイルからローカライゼーションチームが使用するため、及びメッセージエンパリングの確認のためのメッセージカタログを生成するステップ；をさらに具備した請求項3記載の方法。

【請求項5】上記マスタメッセージファイルに、マスタメッセージファイルエントリの種々異なる部分を定義するための導入文字を含むテンプレートを含めるステップ；をさらに具備した請求項3記載の方法。

【請求項6】上記の指定されたアレイ要素に含まれるシンボリックハンドルを用いるステップが：該指定されたアレイ要素に含まれるシンボリックハンドルから導出された情報を用いてHTMLリンクを形成するステップと；該HTMLリンクを必要メッセージ情報と共に表示するステップと；をさらに具備する請求項3記載の方法。

【請求項7】全てのメッセージ用のテキストを保持し、1カ所で情報をサポートすると共に、各々シンボリックハンドル、メッセージテキスト、及びヘルプメッセージテキストを含む複数のメッセージエントリを有する单一のマスタメッセージファイルを設けるステップと；該マスタメッセージファイルからコンパイラヘッダファイルを生成するステップで、該コンパイラヘッダファイルが、マスタメッセージファイル中のエントリのシンボリックハンドルを列挙する列挙ヘッダファイル、及び列挙シンボリックハンドルによって指示されるアレイ要素を有するメッセージアレイファイルを含み、該アレイ要素の各々がアレイをアクセスするために用いられるシンボリックハンドルを含むマスタメッセージファイルエントリからの情報を含んでいるステップと；該列挙ヘッダファイルを用いて、表示しようとする必要メッセージ情報

の指示インデックス用列挙シンボリックハンドルを生成するステップと；該指示インデックス用列挙シンボリックハンドルを用いて該メッセージアレイファイル中の該必要メッセージ情報を含む指定されたアレイ要素にアクセスするステップと；該指定されたアレイ要素に含まれる必要メッセージ情報を表示するステップと；各々ヘルプメッセージ及びヘルプメッセージIDを含む複数のエントリを有するヘルプボリュームを該マスタメッセージファイルから生成するステップと；該指定されたアレイ要素に含まれるシンボリックハンドルを用いて、該必要メッセージ情報と関連した必要ヘルプメッセージテキストを保持する該ヘルプボリューム中のエントリを指定するステップと；ユーザ入力に応答して該必要ヘルプメッセージテキストを選択的に表示するステップと；を具備したコンピュータ実装されるメッセージ管理方法。

【請求項8】全てのメッセージ用のテキストを保持し、1カ所で情報をサポートすると共に、各々シンボリックハンドル、メッセージテキスト、及びヘルプメッセージテキストを含む複数のメッセージエントリを有する單一のマスタメッセージファイルに記憶された情報に基づいてメッセージを管理するためのコンピュータ可読コードが実装されたコンピュータ使用可能媒体；を具備したメッセージ管理装置において：コンピュータに該マスタメッセージファイルからコンパイラヘッダファイルを生成させるよう構成されたコンピュータ可読プログラムコード手段で、該コンパイラヘッダファイルが、マスタメッセージファイル中のエントリのシンボリックハンドルを列挙する列挙ヘッダファイル、及び列挙シンボリックハンドルによって指示されるアレイ要素を有するメッセージアレイファイルを含み、該アレイ要素の各々がアレイをアクセスするために用いられるシンボリックハンドルを含むマスタメッセージファイルエントリからの情報を含んでいるコンピュータ可読プログラムコード手段と；コンピュータに該列挙ヘッダファイルを用いて、表示しようとする必要メッセージ情報の指示インデックス用列挙シンボリックハンドルを生成させるよう構成されたコンピュータ可読プログラムコード手段と；コンピュータに、該指示インデックス用列挙シンボリックハンドルを用いて該メッセージアレイファイル中の該必要メッセージ情報を含む指定されたアレイ要素にアクセスせするコンピュータ可読プログラムコード手段と；コンピュータに該指定されたアレイ要素に含まれる必要メッセージ情報を表示させるよう構成されたコンピュータ可読プログラムコード手段と；コンピュータに各々ヘルプメッセージ及びヘルプメッセージIDを含む複数のエントリを有するヘルプボリュームを該マスタメッセージファイルから生成するよう構成されたコンピュータ可読プログラムコード手段と；コンピュータに該指定されたアレイ要素に含まれるシンボリックハンドルを用いて、該必要メッセージ情報と関連した必要ヘルプメッセージテキストを保持する該ヘルプボリューム中のエントリを指定し；ユーザ入力に応答して該必要ヘルプメッセージテキストを選択的に表示する；よう構成されていることによって該マスタメッセージファイル

ストを保持する該ヘルプボリューム中のエントリを指定させるよう構成されたコンピュータ可読プログラムコード手段と；コンピュータにユーザ入力に応答して該必要ヘルプメッセージテキストを選択的に表示させるよう構成されたコンピュータ可読プログラムコード手段と；をさらに具備したメッセージ管理装置。

【請求項9】コンピュータに該マスタメッセージファイルからローカライゼーションチームが使用するため、及びメッセージナンバリングの確認のためのメッセージカタログを生成させるよう構成されたコンピュータ可読プログラムコード手段；をさらに具備した請求項8記載のメッセージ管理装置。

【請求項10】コンピュータに該指定されたアレイ要素に含まれるシンボリックハンドルから導出された情報を用いてHTMLリンクを形成させるよう構成されたコンピュータ可読プログラムコード手段と；コンピュータに該HTMLリンクを必要メッセージ情報と共に表示させるよう構成されたコンピュータ可読プログラムコード手段と；をさらに具備した請求項8記載のメッセージ管理装置。

【請求項11】全てのメッセージ用のテキストを保持し、1カ所で情報をサポートすると共に、各々シンボリックハンドル、メッセージテキスト、及びヘルプメッセージテキストを含む複数のメッセージエントリを有する單一のマスタメッセージファイルを記憶するコンピュータ可読記憶媒体と；ユーザにメッセージ及びヘルプメッセージを表示するためのディスプレイと；該マスタメッセージファイルからコンパイラヘッダファイルを生成し、該コンパイラヘッダファイルが、マスタメッセージファイル中のエントリのシンボリックハンドルを列挙する列挙ヘッダファイル、及び列挙シンボリックハンドルによって指示されるアレイ要素を有するメッセージアレイファイルを含み、該アレイ要素の各々がアレイをアクセスするために用いられるシンボリックハンドルを含むマスタメッセージファイルエントリからの情報を含んでおり；該列挙ヘッダファイルを用いて、表示しようとする必要メッセージ情報の指示インデックス用列挙シンボリックハンドルを生成し；該指示インデックス用列挙シンボリックハンドルを用いて該メッセージアレイファイル

40 中の該必要メッセージ情報を含む指定されたアレイ要素にアクセスし；該指定されたアレイ要素に含まれる必要メッセージ情報を表示し；各々ヘルプメッセージ及びヘルプメッセージIDを含む複数のエントリを有するヘルプボリュームを該マスタメッセージファイルから生成し；該指定されたアレイ要素に含まれるシンボリックハンドルを用いて、該必要メッセージ情報と関連した必要ヘルプメッセージテキストを保持する該ヘルプボリューム中のエントリを指定し；ユーザ入力に応答して該必要ヘルプメッセージテキストを選択的に表示する；よう構成されていることによって該マスタメッセージファイル

を処理すると共に、メッセージ及びヘルプテキストを表示するよう構成された処理システムと；を具備したメッセージ管理システム。

【請求項12】全てのオプション用のテキストを保持し、1カ所で情報をサポートすると共に、各シンボリックハンドル、オプションテキスト、及びオプションヘルプテキストを含む複数のオプションエントリを有する单一のマスタオプションファイルに基づくコンピュータ実装されるオプション管理方法において：該マスタオプションファイル中の情報に基づいて、オンラインヘルプを生成するために用いられるヘルプボリュームを生成するステップと；該マスタオプションファイル中の情報に基づいて、オプションテキストを生成するために用いられるソースファイルに入れられるヘッダファイルを生成するステップと；該マスタメッセージファイル中の該情報に基づいて、オプションテキストをローカライズするために用いられるオプションカタログファイルを生成するステップと；を具備したオプション管理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、広義には、メッセージシステムの編成及び保守に関し、特に、オンラインヘルプサブシステムを含むメッセージシステムに関する。

【0002】

【従来の技術】たとえばコンパイラやその他のディベロッパ製品を含む多くのソフトウェアシステムは、メッセージドキュメンテーションを含んでいる。たとえば、多くのCコンパイラは、エラーメッセージ及び他の種類のメッセージを生成するデバッグツールを含んでいる。

【0003】コンパイラドキュメンテーションのようなソフトウェアドキュメンテーションを書く際の難題の1つは、ユーザがそれらについての情報を欲するオプションやメッセージが多数あることである。数100のオプションやメッセージのドキュメンテーションをリリースの度に確実にコンパイラソースコードに対して最新の状態に保つことに加えて、作成者は、同じマテリアルが2カ所以上の部分でドキュメント化される場合、ドキュメンテーション上の問題に直面することになる。

【0004】たとえば、いくつかのコンパイラでは、メッセージはユーザガイド、クイックレファレンスドキュメント、及びASCII READMEファイルでドキュメント化される。従って、変更を行う場合、これらの全ての部分で一貫して行わなければならない。

【0005】さらに、国際的な市場を有する製品の場合、メッセージドキュメンテーションは、ユーザがドキュメンテーションをそれぞれの母国語で読むことができるようローカライズしなければならない。従って、ドキュメンテーションの各ソースを個々にローカライズし

なければならぬ。

【0006】

【発明が解決しようとする課題】ソフトウェアのローカライズシステムは当業者には周知である。たとえば、米国特許第5, 513, 342号、5, 243, 519号、5, 148, 541号には、ソフトウェアをローカライズするための方法が開示されている。このようなローカライズのためのシステムは周知であるが、種々のドキュメンテーションコンポーネント間で一貫性を保つという課題は、依然として解決されていない。

【0007】さらに、コンパイラのようなソフトウェアは、複数の異なるプラットフォーム上で用いられるので、メッセージシステムはプラットフォーム間で移植性を持つことが要望される。

【0008】

【課題を解決するための手段】本発明によれば、マスタファイルの内容をコンパイラソースコードに対してリリースの度に最新の状態に保つことができ、メンテナンスは1つのマスタファイルについてだけ行えばよいようにして、メッセージテキスト及びサポート情報がマスタファイルに記憶される。ヘルプメッセージ、エラーメッセージ、及びその他のメッセージはこのマスタファイルから生成される。

【0009】本発明の一つの態様によれば、マスタメッセージファイルは、各メッセージ毎にエントリを有し、各エントリはシンボリックハンドルを含む。マスタメッセージファイルから生成されるファイルは、これらのシンボリックハンドルを用いてマスク情報を編成する。マスタメッセージファイルから生成されるファイルには、たとえば、メッセージ処理インターフェースに含まれるコンパイラヘッダファイル、コンパイラによって出されるエラーメッセージをローカライズするために使用されるメッセージファイル、及びオンラインヘルプボリュームが含まれる。メッセージシステムが更新される場合、マスタメッセージファイルのみが編集され、出力ファイルはその1つのマスタメッセージファイルのみから生成される。

【0010】本発明の他の態様によれば、コンパイラオプションはマスタファイルに記憶され、ヘルプメッセージを生成するのに必要なその他のファイルはマスタファイルから導出される。

【0011】その他の本発明の特徴及び長所については、以下の詳細な説明及び添付図面より明らかとなる。

【0012】

【発明の実施の形態】まず最初に、以下の詳細な説明で使用する用語の定義をまとめて掲げておく。

コマンドラインインタフェース　コンピュータ及びテキストコマンドを用いてプログラムと対話するそのユ

ーザとのインターフェース。

CATGETS () 関数の引数として与えられた数によって指示されるカタログのエントリにアクセスするC関数。

GUI 表示されたアイコンを用いてコマンド及び操作を表すインターフェースであるグラフィカルユーザインターフェースの(Graphical User Interface)の略。

CDE 多くのプログラム用として用いられる共通ウィンドウである共通デスクトップ環境 (Common Desktop Environment) の略。実施例においては、CDEヘルプボリュームによってオンラインヘルプが得られ、ヘルプメッセージはCDEによって生成されるヘルプウィンドウ中に表示される。CDEは、SGML (標準一般化マーク付け言語; Standard Generalized Markup Language) の一形式であるHelp Tag フォーマットでドキュメントを利用する。これらは、CDEドキュメンテーショングループ

(アイ・ビー・エム (IBM) 、ヒューレット・パッカード (HEWLETT-PACKARD) 、サンソフト (SUNSOFT) 、ノベル (NOVELL)) による共通デスクトップ環境1.0プログラマーズ・ガイド

(アディソン・ウェズレー・パブリッシング・カンパニー、米国マサチューセッツ州リーディング (Addison-Wesley Publishing Co., Reading, MA) 、1995年刊) のような各コンピュータシステム用のレファレンスマニュアルに記載されている標準的な技術である。

IPE エディタ、デバッガ及び性能分析ルーチンを含むソフトウェア開発ツールである統合化プログラミング環境 (Integrated Programming Environment) の略。

【0013】図1は、本発明を実施するのに好適なクライアントコンピュータシステム210をブロック図形式で示したものである。図示のクライアントコンピュータシステム210はバス212を有し、バス212は中央処理装置214、システムメモリ216(通常RAM)、入出力(I/O)コントローラ218、ディスプレイアダプタ226を介して接続された表示装置224のような外部機器、シリアルポート228及び230、キーボード232、固定ディスクドライブ234及びフロッピディスク238を装着して動作するフロッピディスクドライブ236、及びCD-ROM242WP装着して動作するCD-ROM装置240のような主要サブシステムを相互接続する。これら以外に、シリアルポート228を介して接続されたマウス244のようなユーザポインティングデバイス及びシリアルポート230を介して接続されたモデム246等、多くの装置を接

続することができる。モデム246は、電話リンクを介してリモートサーバと直接接続することもできれば、POP(ポイント・オブ・プリゼンス (point of presence))を介してインターネットに接続することもできる。あるいは、他の何らかの形態のネットワークインターフェースシステム(図示省略)を用いることもできる。

【0014】同様にして、他の多くの装置またはサブシステム(図示省略)を接続することも可能である。また、以下に説明するように、本発明を実施する上において、必ずしも図1に示す全ての装置がなければならないわけではない。図1に示すようなコンピュータシステムの動作については、当技術分野において周知であり、本願においては詳細な説明は省略する。本発明を実施するためのコードは、システムメモリ216に使用可能または動作可能に配置することもできるし、固定ディスク234、フロッピディスク238、あるいはCD-ROM242のような記憶媒体に記憶することも可能である。

【0015】本発明の実施例においては、コマンドラインインターフェース及び統合化プログラミング環境(IPE)をどちらも有するコンバイラの場合についてメッセージ管理システムを実施する。

【0016】図2は、本発明の一実施例のファイル構造の概要を示す概略図である。図2において、マスタメッセージファイル10は、技術者でもテクニカルライターでも編集することができるASCIIファイルである。マスタメッセージファイル10は、システムの各々のメッセージについてエントリを有する。

【0017】本発明の図示実施例においては、マスタメッセージファイルは、全てのファイルの最新バージョンが維持される共通作業領域であるコンバイラの統合作業領域に保持される。マスタメッセージファイルに対して作業を行う他の技術者やライターは、多くのプログラマの作業を調整するサン・マイクロシステムズ (Sun Microsystems, Inc.)社により製造、配布されているTeamWareのようなプログラムを用いて統合作業領域からマスタメッセージファイルにアクセスする。技術者は、アクセス下マスタメッセージファイルのコピーを自己のワークステーションの子作業領域で編集する。各技術者は、1つのファイルのいくつかのバージョンをオンラインで保持することからしばしば生じる混乱を避けるためにドキュメントの各修正を追跡しやすくするUNIXソースコード制御システム(SCCS)を使用する。SCCSは、プログラムまたはドキュメントが改善または強化されているが、最初の(あるいは前のいづれかの)バージョンが依然として必要とされるような場合に、特に役に立つ。

【0018】マスタメッセージファイルに対して作業する技術者は、以下にさらに詳しく説明するように、メ

セージのテキストをマスタメッセージファイルに直接書き込み、その技術者が子作業領域でのマスタメッセージファイルに対する作業を終了すると、その編集されたバージョンはコンバイラの統合作業領域に戻される。従って、統合作業領域に保持されているマスタメッセージファイルによって、矛盾のない首尾一貫したメッセージシステムを維持するための唯一のドキュメントが得られる。

【0019】図2において、マスタメッセージファイル10にも続く出力ファイルは、コンバイラによって用いられるヘッダファイル12、コンバイラによって出されるエラーメッセージをローカライズするために用いられるメッセージファイル14、及びオンラインヘルプボリューム16である。

【0020】マスタメッセージファイル10の出力は、一組のツールを用いて構築されるが、これらのツールは、この実施例においては、図1に示すコンピュータシステムの種々のコンピュータ使用可能記憶媒体に記憶されるコードとして実施される。これらの出力ファイルは、マスタメッセージファイル10から生成されるファイルの最新バージョンを生成するためのコンバイラの統合ビルト時に、これらのツールによって生成される。統合ビルトは、ファイルを更新するために周期的に行われる。ある時点で、コンバイラのバージョンはカスタマーに対してリリースされる。

【0021】図3に示す第1の処理ステップで、PREPERRツールはマスタメッセージファイル10に対して作業を行い、ヘッダ(.h)ファイル12、オンラインヘルプソース(.htg)ファイル16、及びメッセージカタログを生成するために用いられるソース(.msg)ファイル14を生成する。

【0022】以下に説明するように、オンラインヘルプファイルには、ヘルプシステムトピックIDとヘルプテキストを含むHelp_Body(図14のmsg_helptag.htm)16(1)が入っており、HelpTagマークアップテキストは、たとえば<newline>50のようなSGML(標準一般化マーク付け言語; Standard Generalized Markup Language)よりなり、IPEデバッガを用いるとき生成されるヘルプウインドウに表示されるヘルプメッセージをフォーマットするためにCDEによって使用される。

【0023】メッセージファイルには、メッセージカタログの生成のためにローカライゼーションチームに送られるメッセージカタログソースファイル(uncheck.ed.msg)が入っている。この実施例においては、メッセージカタログは、各メッセージエントリに1つの数が対応付けられたバイナリファイルに変換される。ローカライゼーションを行うときは、コンバイラメッセージのローカライゼーションが行われるように、コンバイラによってCATGETS()が用いられる。

【0024】コンバイラヘッダファイルには、メッセージに対するシンボリックハンドルの列挙目録12(1)

(図6のmsgno.h、及び列挙されたシンボリックハンドルによって指示されるメッセージに関する全ての情報)を含むアレイ12(2)(図7のmsgtxt.h)が入っている。

【0025】PREPERRがマスタメッセージファイル10を処理するためには、各エントリはそのメッセージ及びその関連情報のためのテンプレートに従って書き込まれる。図4のA及びBは、それぞれ、テンプレート及びテンプレート「導入」文字の機能を説明した表である。

【0026】図4のAにおいて、導入文字“-”に対しては、重大度、#msgno、シンボリックリンクまたはハンドル、及び簡略エラーテキストの4つのフィールドがある。“-[”に対しては、ヘルプメッセージとして表示される代替メッセージテキスト41が書き込まれる。

【0027】図5にマスタメッセージファイル10のエントリの一例を示してある。重大度フィールド42は“1”であり、#msgno44フィールドは“#258”であり、シンボリックリンク46は“E_UNDEFINE_SYMBOL”で、マスタメッセージファイルに対するシンボリックハンドルとして用いられ、簡略エラーテキスト458は“未定義シンボル：%s”である。<>で囲まれたテキストは、HelpTagマークアップテキストで、マスタメッセージファイル10にエントリを生成するとき書き込まれる。HelpTagマークアップテキストは、たとえば<newline>50のようなSGML(標準一般化マーク付け言語; Standard Generalized Markup Language)よりなり、IPEデバッガを用いるとき生成されるヘルプウインドウに表示されるヘルプメッセージをフォーマットするためにCDEによって使用される。

【0028】上に述べたように、マスタメッセージファイル10にエントリを加えるためのテンプレートは、ライターまたは技術者がそのエントリを指示するシンボリックハンドルとして機能するシンボリックリンクを入れる必要がある。リリースの都度メッセージナンバリングを維持するために、メッセージに自動的に付番すると共に、メッセージを再使用したりあるいは再度付番したりしないことを規定するリナンバリング規則に全く違反することがないようにするためのツールが用いられる。

【0029】この実施例においては、マスタメッセージファイル10に対してメッセージ番号割り当てツール(asstmsgno)を用いることにより、永久メッセージ番号が自動的に割り当てられる。コンバイラ技術者がメッセージを追加するとき、そのメッセージは、シンボリックハンドルだけで、永久番号なしで追加される。リリース直前に、マスタメッセージファイル10に足してメッセージ番号割り当てツール(asstig

`n m s g n o`) を実行して、メッセージカタログで列挙ヘッダ12(1)を生成するための永久番号が割り当てられるが、これについては、以下に図6を参照してさらに詳しく説明する。

【0030】また、本発明の実施例においては、コンパイラソースコードの新バージョンで新しいメッセージカタログをリリースするとき、比較ツール(`c o m m s g`)が用いられる。この比較ツール(`c o m m s g`)は、メッセージカタログを前のリリースのメッセージカタログと比較して、リナンバリング規則に違反していないかどうかの確認が行われる。

【0031】図6及び7にコンパイラヘッダファイル12を例示してある。図6に示す列挙ヘッダ12(1)は、メッセージに対するシンボリックハンドルの列挙リストが書き込まれる。周知のように、列挙シンボリックハンドルは整数と同様に作用する。従って、シンボリックハンドルを列挙する機能は、マスタメッセージファイルのエントリに割り当てられるメッセージ番号(`m s g n o`)として機能する整数と等価の列挙シンボリックハンドルを生成する。

【0032】図7に示すメッセージアレイ12(2)は、メッセージ情報のアレイが書き込まれ、アレイ中の各要素には列挙シンボリックハンドル(`m s g n o`)によってインデックスが付される。アレイの各エントリは、`C A T G E T S <>`に関するメッセージのテキスト60、シンボリックハンドル62、ヘルプシステムトピックID64、及びそのメッセージ番号66を含む。メッセージ番号は列挙ヘッダファイル12(1)によって生成される列挙シンボリックハンドルである。ヘルプシステムトピックID64は、この実施例においてはヘルプボリュームは下線文字を扱うことができず、ケースインセンシティブであるため、必要である。シンボリックハンドル62は、従って、ヘルプシステムトピックID64に変換される。ヘッダファイルは、任意のメッセージについてコンパイラが必要とする全ての情報が書き込まれる。

【0033】本発明のこの実施例でマスタメッセージファイル10から生成されるヘッダファイル12に記憶されたメッセージに関する情報をアクセスするために使用されるコンパイルインタフェースルーチンの定義を図8に示してある。これらのインタフェースルーチンは、やはりメッセージアレイ12(2)を含む単一のメッセージソースファイルで定義される。これらの各インタフェースルーチンは、列挙シンボリックハンドルが`m s g n o`として機能するインタフェースルーチンに対する引数として与えられるメッセージ番号(`m s g n o`)によって指示されるアレイ要素から情報を返す。たとえば、ルーチン`g e t _ m s g (i n t m s g n o)`は、`m s g n o`によって指示されるアレイ要素からメッセージテキストを返す。ルーチン`g e t _ h e l p t a g (i n`

`t m s g n o`)は、`m s g n o`によって指示されるアレイ要素からヘルプタグを返す。このヘルプタグは、CDEによってヘルプボリューム16でしいヘルプメッセージにアクセスするために用いられる。このように、コンパイラがメッセージシステムとのインターフェースを取るのに、単一のメッセージソースファイルしか用いられない。

【0034】図9は、エラーメッセージを生成し、表示するための手順を示すフローチャートである。上に述べたように、エントリは、マスタメッセージファイル10に書き込まれる。マスタメッセージファイル10をリリースする前に、PREPERRツールがヘッダファイル12を生成し、コンパイラはこれを用いてエラーメッセージ及びオンラインヘルプメッセージへのリンクを生成する。プログラムをコンパイラする際にエラーが生じると、コンパイラは列挙ヘッダファイル12(1)を用いて、エラーメッセージを含むアレイ要素を指示する列挙シンボリックハンドルを生成する。列挙シンボリックハンドルは、その後ユーザに表示されるエラーメッセージを生成するのに必要な全ての情報を含むアレイ要素にアクセスするために用いられる。

【0035】本発明のこの実施例においては、アレイヘッダ12(2)にヘルプシステムトピックIDが入れられ、コンパイラによって生成されるHTML(図12)に入れられるヘルプテキストメッセージへのアンカーリンクのHREF属性によって指定されるファイル名の一部として用いられる。上に述べたように、この実施例においては、ヘルプボリュームは下線文字を扱うことができず、ケースインセンシティブであるため、ヘルプシステムトピックIDが必要である。そのために、シンボリックハンドル62はヘルプシステムトピックID64に変換される。

【0036】次に、ヘルプ出力ファイル構造を示す図10及び図10に示すファイルを生成するためのステップを示す図11のフローチャートを参照して、オンラインヘルプファイルの生成について説明する。コンパイラ用のヘルプファイルは、下記によって構成される。

1. IPEの一部をなし、HTMLファイルを表示するBuild Outputウィンドウにエラーメッセージに関するヘルプを表示するために用いられるヘルプボリューム50
2. それらと同じエラーメッセージのASCIIテキストファイル。これは、IPEを使わないことを選択した場合、あるいはコマンドラインインターフェースを使用しなければならない場合に利用可能である。

【0037】まずBOW環境を念頭において図10及び11を参照しつつ説明すると、ヘルプボリューム50を構築するためにHelp Body16(1)及びHelpHeader16(2)に対して作業するヘルプコ

ンバイラ `d t h e l p t a g 5 2` を用いてヘルプボリューム (`SUNW_PRO_SC_error.hel p.s d l`) 5 0 が構築される。

【0038】ユーザが I P E を用いて B O W に表示されたエラーメッセージに関するオンラインヘルプにアクセスするために、ヘルプエラーメッセージへのリンクが設けられる。エラーメッセージは、マスターメッセージファイル 1 0 から生成されるヘッダファイル 1 2 に入れられる情報から B O W で使用するようにコンパイラによって HTML で生成される。

【0039】図 1 2 はコンパイラメッセージ HTML の一例を示し、図 1 3 は、B O W 1 2 0 中に表示される図 1 1 の HTML 及び I P E エラーメッセージヘルプウィンドウ 1 2 2 に示されるエラーメッセージを示したものである。ヘルプボリューム 5 0 からエラーメッセージへのリンクはアレイヘッダファイル 1 2 (2) に記憶された情報を用いて構築される。

【0040】ユーザが `B u i l d _ O u t p u t` 「ウィンドウでプログラムをコンパイラすると、全てのエラーはそのウィンドウ 1 2 0 に表示される。エラーメッセージの下線が付されたテキスト上でクリックすると、そのエラーを含むソースコードの行が表示される。情報グリフ (アイコン) 1 2 4 をクリックすると、I P E エラーメッセージヘルプウィンドウ 1 2 2 にエラーメッセージのためのオンラインヘルプが表示される。この場合、エラーメッセージは、メッセージテンプレートに書き込まれる代替テキストであるということに留意すべきである。

【0041】図 1 0 に戻って、オンライン A S C I I ファイル (`SUNW_S_P R O _ S C_error.hel p.t x t`) 5 4 は、`H e l p_B o d y` ファイル 1 6 (1) からたとえば図 1 4 の `< n e w l i n e >` のような `H e l p_T a g` マークアップを取り去ってストリップドファイル (`m s g_h e l p.h t g.s t r i p h t`) 5 8 を生成するスクリプト (`s t r i p h t`) 5 6 を介して生成される。たとえば、A S C I I ファイル 5 4 は、ファイルを紹介し、より一貫性のあるものにするために、何らかのヘッダテキスト (`m s g_h e l p.p_r e a s c i i.t x t`) が付加される。

【0042】`H e l p_B o d y` 1 6 (1) 及び `H e l p_H e a d e r` 1 6 (2) に書き込まれるファイルの例をそれぞれ図 1 4 及び 1 5 に示してある。図 1 6 には、図 1 4 に示す `H e l p_B o d y` のストリップドバージョンである A S C I I ファイルヘルプが示されている。

【0043】この実施例においては、本願で説明したメッセージ管理システムは図 1 のコンピュータシステム 2 1 0 に実装される。

【0044】以上、本発明をその実施例に基づき説明した。当事者にとっては、本発明の変更態様や代替態様は

自明であろう。たとえば、ヘルプボリュームでは、ヘルプメッセージを表示するために用いられるインターフェースによって、`H e l p_T a g` ではなく HTML を用いることもできる。さらに、メッセージを管理する場合について上に説明したのと同じ方法をコンパイラオプションを管理するために使用することも可能である。それらのオプションはマスタファイルに保持し、他の全てのファイルをマスタファイルから導出するようにしてよい。従って、本願の説明、記載は、特許請求の範囲に記載する場合を除き、本発明を限定するためのものではない。

【0045】

【発明の効果】本発明によれば、メッセージシステムの改訂及び更新がただ一つのファイルを編集するだけで行われ、ソフトウェアをローカライズなどにおいて、種々のドキュメンテーションコンポーネント間で一貫性を確保することができる。

【図面の簡単な説明】

【図 1】本発明を実施するのに好適なコンピュータシステムを示すブロック図である。

20 【図 2】マスターメッセージファイルから生成される出力ファイルを示す説明図である。

【図 3】マスターメッセージファイルの初期処理を示す説明図である。

【図 4】4 A はマスターメッセージファイルエントリのテンプレートを示し、4 B は主要文字とフィールド記述の表を示す。

【図 5】マスターメッセージファイルエントリを示す説明図である。

30 【図 6】コンパイラ列挙ヘッダファイルの一部を示す説明図である。

【図 7】コンパイラメッセージアレイヘッダファイルの一部を示す説明図である。

【図 8】メッセージハンドリングインターフェースルーチンを示す説明図である。

【図 9】メッセージを生成するステップを示すフローチャートである。

【図 1 0】オンラインヘルプファイルの生成過程を示す説明図である。

40 【図 1 1】ヘルプメッセージを生成するステップを示すフローチャートである。

【図 1 2】C コンパイラエラーメッセージ HTML を示す説明図である。

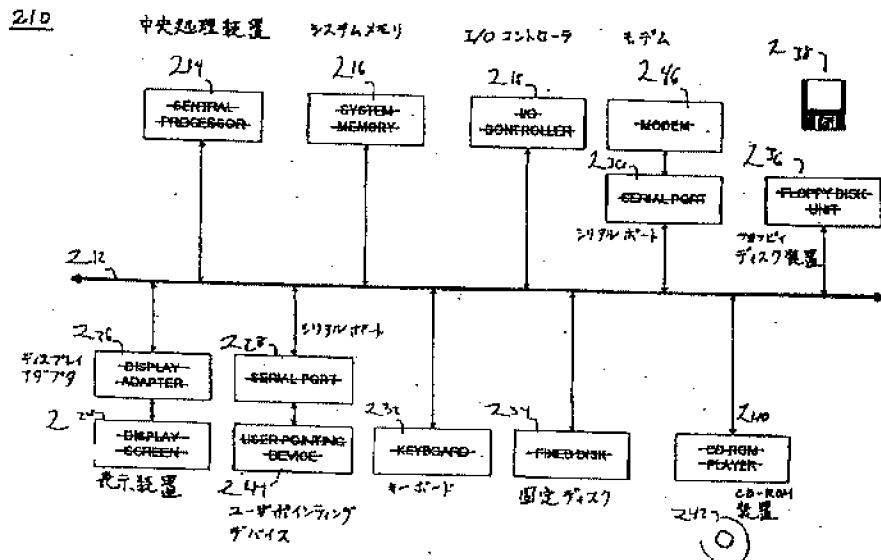
【図 1 3】ビルド出力ウィンドウ及びエラーメッセージウィンドウを示す説明図である。

【図 1 4】`H e l p_B o d y` ファイルの一部を示す説明図である。

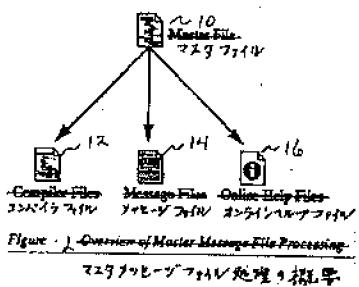
【図 1 5】`H e l p_H e a d e r` ファイルの一部を示す説明図である。

【図 1 6】A S C I I ファイルヘルプメッセージを示す説明図である。

[1]



[図2]



[図3]

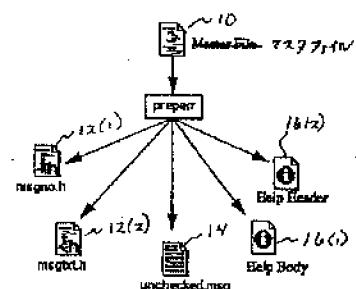


圖 5-1

~~- C:\PROG\X\DIRECTX_11\SAMPLE undefined symbol is
-| undefined symbol
-| covariance ~ 50
You have referred to a symbol for which there is no
declaration in scope.~~

[6]

179

```

j18n.c defines these support routines:
#include "mconfig.h"

extern void j18n_initchar("char"); /* Find source file path
/* */

extern void tidy_fini();

extern void *get_symbolic_handle(); /*

extern const char *get_symbolic_name();
extern const char *get_file_cstring(const char *name);
extern const char *get_hex_ltag(const char *tag);
extern const char *get_hex_ipname(const char *tag);
extern const char *find_hex_label(const char *tag, const
        char *symbolic); /*

/* ... var symbolic handle or help system topic id */

extern int get_candid(char*); /* For License library
*/



j18n.h declares these macros and the above functions

#define FILE_TYPE_AB
#define DEFNAME_DEFNAME_AB
#define DEFNAME_DEFNAME_C
#define DEFNAME_DEFNAME_D
#define DEFNAME_DEFNAME_E
#define DEFNAME_DEFNAME_F
#define DEFNAME_DEFNAME_G
#define DEFNAME_DEFNAME_H
#define DEFNAME_DEFNAME_I
#define DEFNAME_DEFNAME_J
#define DEFNAME_DEFNAME_K
#define DEFNAME_DEFNAME_L
#define DEFNAME_DEFNAME_M
#define DEFNAME_DEFNAME_N
#define DEFNAME_DEFNAME_O
#define DEFNAME_DEFNAME_P
#define DEFNAME_DEFNAME_Q
#define DEFNAME_DEFNAME_R
#define DEFNAME_DEFNAME_S
#define DEFNAME_DEFNAME_T
#define DEFNAME_DEFNAME_U
#define DEFNAME_DEFNAME_V
#define DEFNAME_DEFNAME_W
#define DEFNAME_DEFNAME_X
#define DEFNAME_DEFNAME_Y
#define DEFNAME_DEFNAME_Z

```

[图 4]

ファイルの最初に書けるテンプレートの例:
The example of the beginning of the file looks like this:

```
-- security /home/john user=brink-werke-cust  
+|- multihost-client-cust-test  
++ exec-explanation-help-cust  
+|- vmbase-cust-cust-test  
+|- conneg  
+|- testscript-path
```

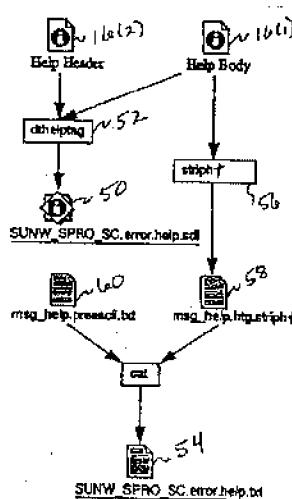
B

Lead-in char	Field Description
-	Normal (standard) error text. Consists of four fields: severity, #megno, symbolic link, and brief-error-text. Severity is a number and is defined as a severity level for use by the compiler. Currently, C does not use this; it is set to 3. #megno is generated automatically by the assignmeigno utility. There are some naming rules for the symbolic links, for example, a link must begin with the letter E, must consist of all uppercase letters, and must be unique. Brief-error-text is the actual text of the compiler message.
!	Alternative text used in the help system for the error message, for example, replacing the print F() directive symbol, like t3
++	Error explanation help text
→	Verbose error text (reserved for future use)
#	Error source file comments
@@	Comment documenting regression test which stresses this message

-Load in Characters and Field Descriptions-

进入文字 * 为所有

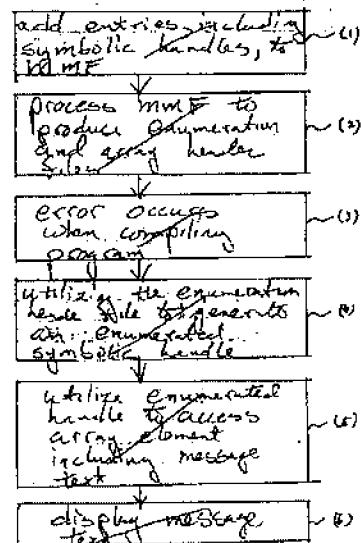
[圖 10]



Production of the Online Help Files

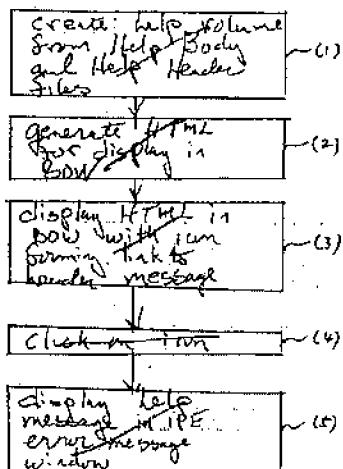
[図7]

[図9]



- (1) シンボリックハンドルを含むエントリをマスタメッセージファイルに書き込む
 - (2) マスクメッセージファイルを処理して列挙及びアレイヘッダファイルを生成する
 - (3) プログラムのコンパイラ時にエラー発生
 - (4) 列挙ヘッダファイルを用いて列挙シンボリックハンドルを生成する
 - (5) 列挙シンボリックハンドルを用いてメッセージテキストを含むアレイ要素にアクセスする
 - (6) メッセージテキストを表示する

[☒ 1 1]



- (1) Help Body及びHelp Headerファイルからヘルプボリュームを生成する
 - (2) BOWに表示するためのHTMLを生成する
 - (3) ヘッダメッセージへのリンクを形成するアイコンによってBOWをHTMLを表示する
 - (4) アイコンをクリックする
 - (5) IPEエラーメッセージワインドウにヘルプメッセージを表示する

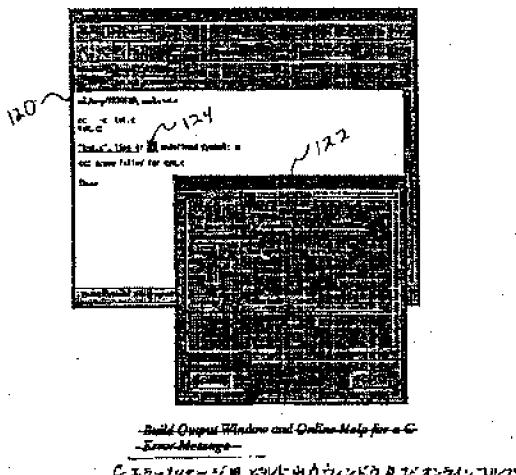
[图12]

アラートメッセージ HTML

```
<library> ScriptResults FILE "msg_help.htm">
<scriptInfo>
<title> Error and Warning Messages
</scriptInfo>
<scriptResults>
<scriptInfo>
<messageTopic>Error and Warning Messages
<listOrder> Right
<list><msg> Click \[LINK\] Help->Topics <msg> [LINK]
</list>

```

[图] 3]



[図14]

```
;;other from 14+1-UNDEFINED-SYMBOLS-the undefined symbols
symbol
undefined ~ 50
You have referred to a symbol for which there is no
declaration in scope.'
```

Example of code that generates the message:

```
(defun
  (x)
  (y))
  )
  (values
```

Message ID: Computer> 1-UNDEFINED-SYMBOL <computer>
 (defun
 ...
)

[圖 1-6-1]

```
undefined symbol

You have referred to a symbol for which there is no
declaration in scope.

Example of code that generates this message:

f(x)id();
g(s);
i

Message ID: E_UNDEFINED_SYMBOL
```

ASCII File Help for a C Error Message

【手続補正書】

【提出日】平成9年11月5日

【手続補正1】

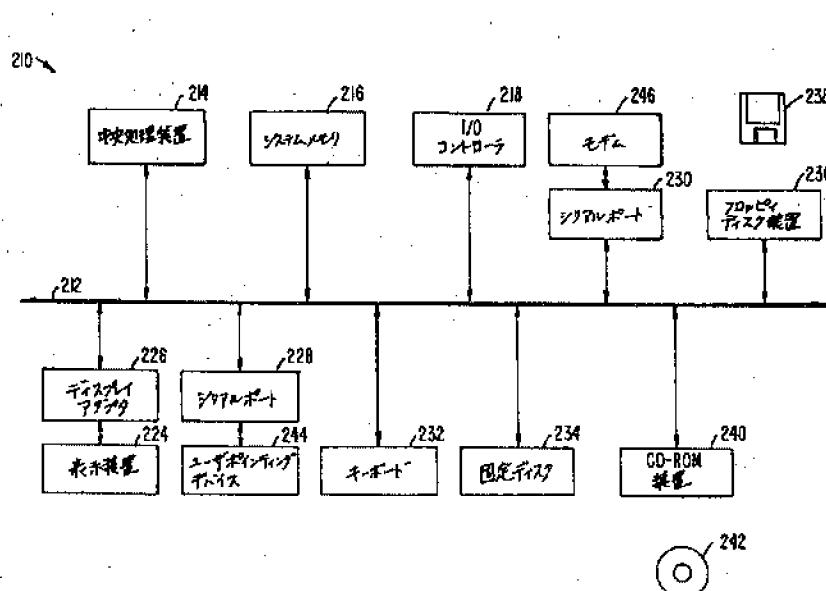
【補正対象書類名】図面

* 【補正対象項目名】全図

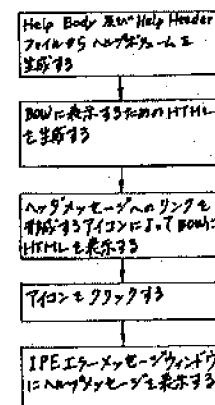
【補正方法】変更

* 【補正内容】

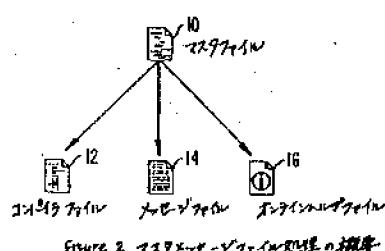
【図1】



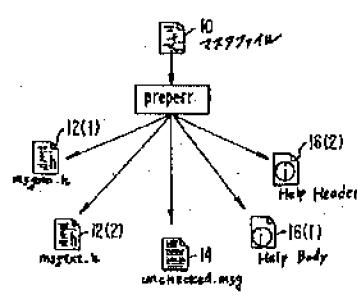
【図11】



【図2】



【図3】



【図7】

```

#ifndef MSGTXT_H
#define MSGTXT_H

/* This file was created automatically from the error
   description file.
   Do not edit this file, edit the error description
   file.

#include <etdddef.h>

struct XBB_error_t {
    const unsigned int severity;
    const int catsetno;
    const int catno;
    const char * tag;
    const char * txt;
    const char * help_tag;
};

static const struct XBB_error_t XBB_error[] = {
    {0, 1, 0, "MESSAGENOTFOUNDINUSED",
     NULL }, /* Error: Message number 0 used. */
    {1, 1, 256, "UNDEFINED_SYMBOL",
     NULL }, /* Lots deleted ... */
    {1, 1, 256, "UNDEFINED_SYMBOL",
     NULL }, /* Lots deleted ... */
    {1, 1, 256, "UNDEFINED_SYMBOL",
     NULL }, /* Lots deleted ... */
    {1, 1, 256, "UNDEFINED_SYMBOL",
     NULL } /* Lots deleted ... */
};

#endif /* MSGTXT_H */

```

【図5】

```

-- 1 #258 E_UNDEFINED_SYMBOL undefined symbol: %s
|[ undefined symbol
++ <newline>~50
You have referred to a symbol for which there is no
declaration in scope.

Example of code that generates the message:
<vex>
f(void)
g(i);
}
</vex>

```

マスターメッセージファイルのエラー例

【図4】

11Bn.c defines these support routines:

```

-- SEVERITY (MSGNO) MAG-TAG BRIEF-ERROR-TEXT
-{ SANITIZED-BRIEF-ERROR-TEXT
++ ERROR-EXPLANATION-HELP-TEXT
-+ VERBOSE-ERROR-TEXT
-# COMMENT
-@ TESTNAME-PATH

```

A

LEAD-IN CHAR	FIELD DESCRIPTION
-	NORMAL (STANDARD) ERROR TEXT. CONSISTS OF FOUR FIELDS: SEVERITY, #MSGNO, SYMBOLIC LINK, AND BRIEF-ERROR-TEXT. SEVERITY IS A NUMBER AND IS DEFINED AS A SEVERITY LEVEL FOR USE BY THE COMPILER. CURRENTLY, C DOES NOT USE THIS; IT IS SET TO 1. #MSGNO IS GENERATED AUTOMATICALLY BY THE ASSIGN MSGNO UTILITY. THERE ARE SOME NAMING RULES FOR THE SYMBOLIC LINKS, FOR EXAMPLE, A LINK MUST BEGIN WITH THE LETTER E, MUST CONSIST OF ALL UPPERCASE LETTERS, AND MUST BE UNIQUE. BRIEF-ERROR-TEXT IS THE ACTUAL TEXT OF THE COMPILER MESSAGE.
-[ALTERNATIVE TEXT USED IN THE HELP SYSTEM FOR THE ERROR MESSAGE, FOR EXAMPLE, REPLACING THE PRINTE() DIRECTIVE SYMBOLS, LIKE %S
++	ERROR EXPLANATION HELP TEXT
-+	VERBOSE ERROR TEXT (RESERVED FOR FUTURE USE)
-#	ERROR SOURCE FILE COMMENTS
-@	COMMENT DOCUMENTING REGRESSION TEST WHICH STRESSES THIS MESSAGE

B

導入文字と7-1-PEL記述

【図6】

```

#ifndef _MSGNO_H
#define _MSGNO_H
/* Error Message Values -
 * This file was created automatically from the error
 * description file.
 * Do not edit this file, edit the error description
 * file:
 *   ../../inc/arc/acomp_msg.txt
 */

typedef enum msgno_enum {
    MESSAGEZEROISNOTUSED = 0,
    /* ... Lots Deleted ... */
    E_UNDEFINED_SYMBOL,
    /* ... Lots Deleted ... */
    MESSAGELASTISNOTUSED
} msgno_t;

```

コンパイラメッセージファイル msgno.h

【図15】

```

<entity ScriptResult FILE "msg_help.htm">
<metainfo>
<title> Error and Warning Messages
<scriptResult>
<metainfo>
<homeTopic>Error and Warning Messages
<list order="tight">
<item><link E_UNDEFINED-SYMBOL Definition>E_UNDEFINED-
SYMBOL</link>...

```

オンラインヘルプソースファイル SUNW_SPRO_SC_error_help.htm

【図8】

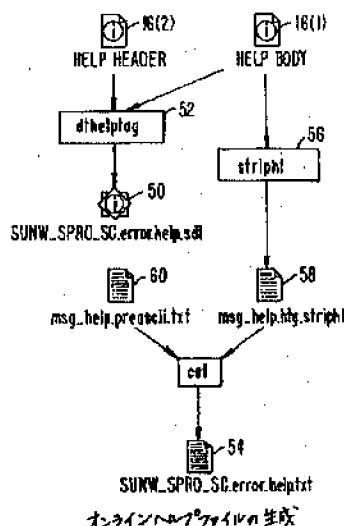
```

11Bn.c defines these support routines:
#include "msgtxt.h"
extern void 11Bn_init(char *path); /* Find Yourself
path */
extern void 11Bn_fini();
/* msgno == symbolic handle */
extern const char * get_mag(int msgno);
extern const char * get_help_tag(int msgno);
extern const char * get_mag_tag(int msgno);
extern unsigned int get_mag_severity(int msgno);
extern msgno_t find_mag(const char * tag, int tagkind);
/* via symbolic handle or help system topic
id */
extern nl_catd get_catd(void); /* For License library
*/
11Bn.h declares these macros and the above functions:
#include <nl_types.h>
#define DEFNOV_SEENO 1
#define GETMSG( MSGNO ) get_mag( MSGNO )
#define MSGNO( MSG ) MSG
#define MSGTAG( MSG ) 1
#define MSGHELPTAG 2

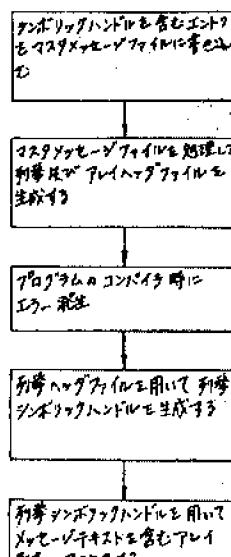
```

7-2-7-Help 6-77-2-7-7-7

【図10】



【図9】

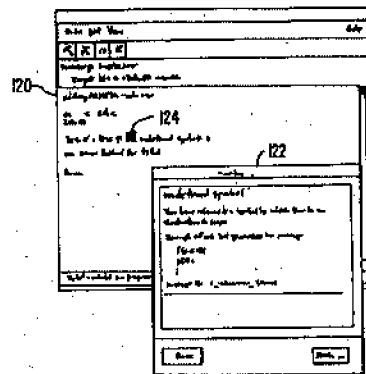


メッセージハンドルを用いてメッセージを登録する

【図12】

```
$ cc tst.c -c
(/tmp/1230510)tst.c:
<HTML>
<PRE>
<A HREF="editor:tst.c:4:0:error:undefined symbol:
a*><tst.c*, line 4</A>: <A HREF="help:/set/dist/sparc-
S2/bin/..SC4.2/bin/..lib/locale/C/help/
SUNW_S2R0_SC.error_help.sdl:E-UNDEFINED-
SYMBOL:undefined symbol: a
*><IMG SRC="SUNW-internal-info-sign" ALT="???"></A>
undefined symbol: a
</PRE>
</HTML>
</HTML>
cc: acomp failed for tst.c
C:\コンピュータ\エーティング\HTML
```

【図13】



【図14】

```
<otherfront id=E-UNDEFINED-SYMBOL><head>undefined
symbol
<newline>~50
You have referred to a symbol for which there is no
declaration in scope.

Example of code that generates the message:
<rex>
f(void){
g(i);
}
</rex>

Message ID: <computer> E_UNDEFINED_SYMBOL </computer>
<newline>
```

オンラインヘルプソースファイル msg-help.htm

【図15】

```
undefined symbol
You have referred to a symbol for which there is no
declaration in scope.

Example of code that generates the message:
<rex>
f(void){
g(i);
}
</rex>

Message ID: E_UNDEFINED_SYMBOL
```

C:\メッセージ用 ASCII ファイル&gt;

フロントページの続き

(71)出願人 591064003

901 SAN ANTONIO ROAD
PALO ALTO, CA 94303, U.
S. A.

(72)発明者 キャサリン・ウォーカー

アメリカ合衆国・95014・カリフォルニア
州・カッパチーノ・ノーススカイ スクエ
ア・10981

(72) 発明者 ロバート・エフ・モリ
アメリカ合衆国・94025・カリフォルニア
州・メンロ パーク・ミドル アヴェニ
ュ・937

